International Journal of Mathematics and Soft Computing
Vol.4, No.1. (2014), 35 - 49.

ISSN Print : 2249 – 3328
ISSN Online: 2319 – 5215

Algorithms to compute some dominating sets of *ES* (*n*, *k*) and *ESC* (*n*, *k*)

S. Pious Missier
Research Department of Mathematics,
V.O. Chidambaram College, Thoothukudi,
Tamilnadu, INDIA.

A. Anto Kinsley
Research Department of Mathematics,
St. Xavier's College, Palayamkottai,
Tamilnadu, INDIA.

Evangeline Prathibha Fernando
Research scholar, Research Department of Mathematics,
V.O. Chidambaram College, Thoothukudi,
Tamilnadu, INDIA.
Email: evaniebrain@gmail.com

Abstract

Dominating sets are used in the allocation of finite resources to massively parallel architectures. In this paper, we propose a new way of determining a minimal dominating set, connected dominating set and total dominating set of an extended star graph *ES* (*n*, *k*) as well as the extended star graph with cross-connections *ESC* (*n*, *k*) using the levels of its architecture instead of nodes thereby reducing the running time of a domination algorithm significantly.

Keywords: Network, dominating set, connected dominating set, total dominating set, star graph, extended star graph.

AMS Subject Classification (2010): 05C85.

1 Introduction

The extended star graph (*ES*) and its variation, the extended star graph with cross-connections (*ESC*) [2] are massively parallel network architectures that have been designed by A. Anto Kinsley et al., in order to overcome the lack of scalability or hierarchy in the star graph network [1]. The extended star graph exhibits improved diameter, constant degree of connectivity and a basic building block. In this paper, we have designed some sequential algorithms to determine some domination parameters of the Extended Star Graph (*ES* (*n*, *k*)) as well as the Extended Star Graph with Cross-Connections (*ESC* (*n*, *k*)) such as minimal dominating set connected dominating set and total dominating set. All the algorithms, except the algorithms to determine a connected dominating set for the *ES* and *ESC* have a linear time complexity. Since the algorithms are level dependent, their time complexities are measured in terms of *k*, which denotes the number of hierarchical levels in the *ES* and *ESC* rather than the number of nodes in *ES*, *ESC*. For graph theoretic terminology we refer to Chartrand and Lesniak [5] and Haynes et al. [7, 8].

2 Extended Star Graph

The extended star graph (*ES*) [2] is a scalable network that takes the advantage of star graph and tree architectures. The extended star graph architecture consists of two types of nodes, the processor elements (*PE*s) and the network controllers (*NC*s). The basic module of the extended star graph, represented by *ES* (*n*, 1) consists of an *n*-star of *PE*s, $P_1, P_2 \dots P_{n!}$ at level zero and one *NC* represented as P_0 at level one. The *PE*s of the *n*-star are connected to the *NC* by links. The *ES* (3, 1) with two hierarchical levels is shown in Figure 1.

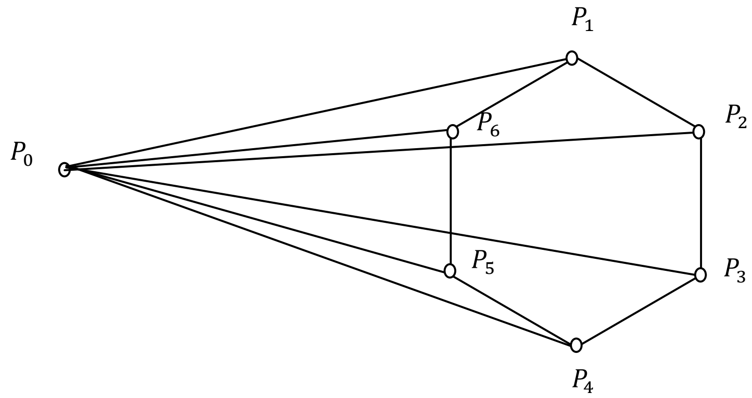

Figure 1: Extended star graph *ES* (3, 1).

The *PEs* are all at the lowest (or zero) level and the *NCs* are at higher levels. Using *n*! such building blocks we can construct an *ES* (*n*, 2) with three levels of hierarchy. The number of hierarchical levels of *ES* (*n*, *k*) is denoted as n_h and its value is *k* + 1.

In general, we represent an extended star graph comprising of *n*- stars and (*k* + 1) hierarchical levels by *ES* (*n*, *k*). The *ES* (*n*, *k*) can be built using *n*! *ES* (*n*, *k* – 1)s. The *PE*s and *NC*s of the *ES* (*n*, *k*) are represented by the combination of permutations $P_1, P_2 \dots P_{n!}$ on *n* symbols 1, 2, 3…. *n*. The combination of two permutations is denoted as $P_i P_j$, simply P_{ij}. Denote the set {1, 2…. *n*!} as Λ.

In the *ES* (*n*, *k*), the *PE*s are addressed by the combination of *k* permutations as $P_{i_1 i_2 \dots i_k}$, where $i_1, i_2 \dots i_k \in \Lambda$. In general, the nodes of an *n*-star at level *j* are represented by combination of (*k* – *j*) permutations, where $0 \le j \le k$.

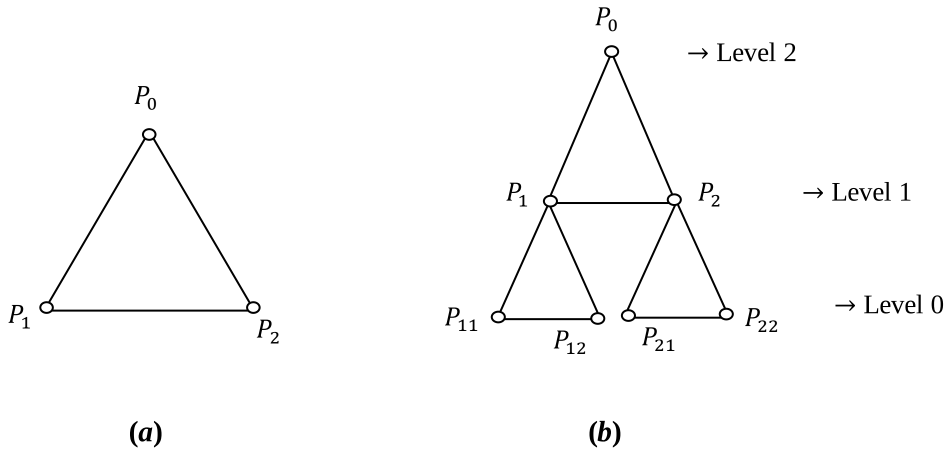

Figure 2(*a*): *ES* (2, 1) **Figure 2(*b*):** *ES* (2, 2)

Thus a node of level *j* is represented by $P_{i_1 i_2 \ldots i_{k-j}}$ and a node at level zero is represented by $P_{i_1 i_2 \ldots i_k}$ where $i_1, i_2 \ldots i_k \in \Lambda$. P_{i_1} indicates the address of the top parent node at level $k - 1$, $P_{i_{k-j-1}}$ corresponds to the parent node and $P_{i_{k-j}}$ represents the position of the node in the *n* - star. The addresses of all pairs of neighbouring *PEs* differ in the last component. Hence the *ES* (*n*, *k*) is constructed by interconnecting *n*! number of *ES* (*n*, *k* – 1) s and the *NC* of level *k* form an *ES* (*n*, *k*). The *ES* (2, 1) and *ES* (2, 2) are shown in Figure 2 (*a*) and (*b*).

3 Extended star graph with cross-connections

The extended star graph with cross-connections [2], *ESC*, is constructed as *ES* (*n*, *k*) and is built using basic building block, each basic building block represented by *ESC* (*n*, *k*) consists of *n*-star of processor elements (*PE*s), a network controller (*NC*), links from each *PE* to the *NC* and additional links for interconnecting with other new basic building blocks as shown in Figure 3. There are cross-links forming a network among the *NC*s and the *PE*s. Also there are two *NC*s with the same address for all levels *j* > 0; $P_{i_1 i_2 \ldots i_{k-j}}$ (*NC*) and $P'_{i_1 i_2 \ldots i_{k-j}}(NC')$ where as each *PE* of level zero represented by $P_{i_1 i_2 \ldots i_{k-1} i_k}$ is connected to two *NC*s of level 1 and level $1'$with the addresses $P_{i_1 i_2 \ldots i_{k-1}}$(*NC*) and $P'_{i_1 i_2 \ldots i_{k-j}}(NC')$ respectively.

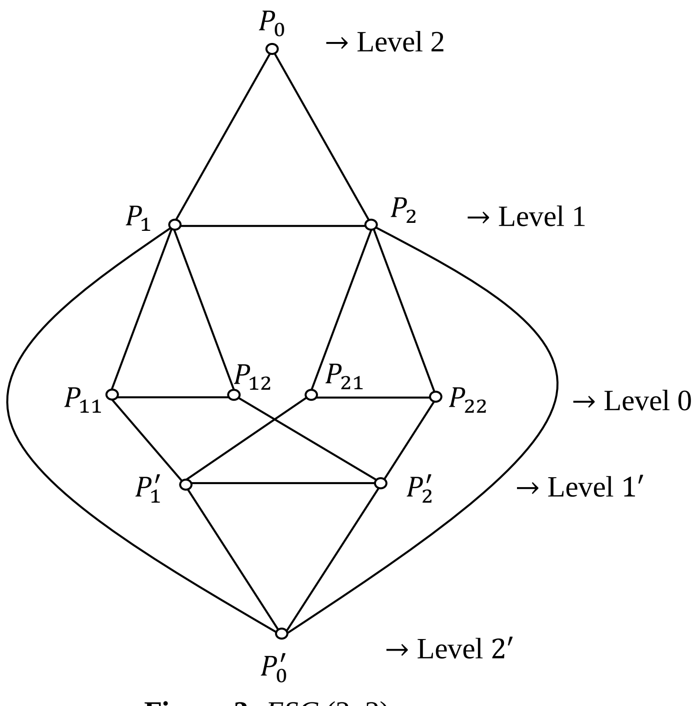

Figure 3: *ESC* (2, 2).

A node *NC* at level *j* (0 < *j* < *k*) represented by $P_{i_1 i_2 \ldots i_{k-j}}$ is connected to two parent nodes $P_{i_1 i_2 \ldots i_{k-j-1}}$ of level *j* – 1 and $P'_{i_1 i_2 \ldots i_{k-j-1}}$ of level $(j-1)'$. In addition, an *NC* at level *j*, 0 < *j* < *k* with address $P_{i_1 i_2 \ldots i_{k-j-1}}$ is connected to *n*! child nodes at level *j* + 1. An *ESC* (*n*, *k*) comprises of $(n!)^{k-1}$ *n* stars of *PE*s at the lowest level of hierarchy and two *NC*s at the highest level of hierarchy. In general there are $2(n!)^{k-j-1}$ n-stars of *NC*s at level *j*. Hence there are $(n!)^{k-1}$ $(n!)$ = $(n!)^k$ *PE*s at the lowest level.

Hence there are three types of links in *ESC* (*n*, *k*). They are (i) the star links connecting nodes of the *n*-stars, (ii) the *ES* links forming a *n*!- ary tree network with *NC*s and *PE*s and (iii) the cross links forming a network among the *NC*s and *PE*s. These links are illustrated in Figure 4.

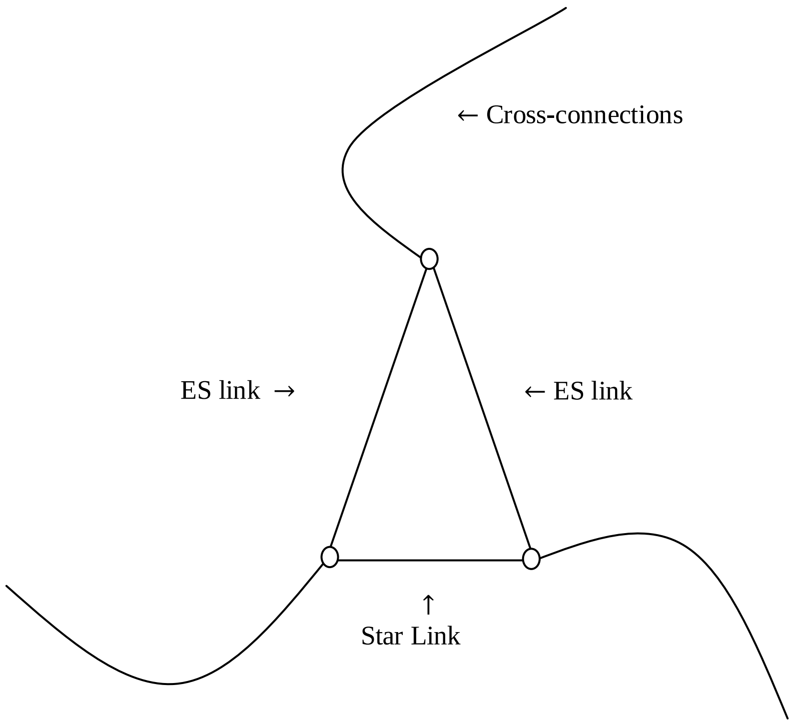

Figure 4: *Basic building block of ESC* (*2*, *k*).

4 Domination parameters of the Extended Star Graph

In this section, we propose algorithms to determine some domination parameters of the extended star graph *ES* (*n*, *k*) and also compute their complexity.

4.1 Minimal dominating set of *ES* (*n, k*)

4.1.1 Design of Algorithm

The following algorithm defines n_h= k + 1 and divides all numerical values of n_h into three mutually independent cases, namely (i) when n_h = 2, 3 (ii) when n_h is of the form 3*m* + 1, *m* = 1,2… and (iii) when n_h is not of the form 3*m* + 1, *m* = 1, 2… which are dealt in Case 1, Case 2 and Case 3 respectively. In Case 1, the dominating set *S* includes all nodes belonging to level 1 of *ES* (*n*, *k*). Case 2 employs a recursive method to include all levels of the form 3*i* + 1, *i* = 0, 1… and culminates at level *k*. Case 3 employs a similar recursive method to include all levels of the form 3*i* + 1, *i* = 0, 1… but it ends with a level of the form 3*i* + 1, that immediately precedes *k*.

4.1.2 Algorithm to find a minimal dominating set of *ES* (*n*, *k*)

```
Begin
Input k
int n_h, i, t ;
n_h = k + 1
S ← φ
if n_h = 2, 3
    Go to Case 1
else
   if  n_h ≡ 1 (mod 3)
    Go to Case 2
else
     Go to Case 3
Case 1:
          S ← S ∪ {All nodes of Level "1"}
 break ;
Case 2:
          S ← S ∪{P_0}
              for i = 0 to k, i = i + 1  do
                  t ←3i + 1;
                    while ( k – t ≥ 2)
                     do
                       S ← S ∪{All nodes of Level "t"}
              break;
Case 3:
          for i = 0 to k, i = i + 1 ;
              t ← 3i + 1;
               while ( k – t ≥  0)
                do
               S ← S ∪{All nodes of Level "t"}
          break;
   Output  S
End
```

Theorem 4.1. Algorithm 4.1.2 determines a minimal dominating set of *ES* (*n*, *k*) and its complexity is O (*k*).

Proof: First we prove that the above algorithm determines a dominating set *S* of *ES* (*n, k*). When we analyze the above algorithm, it is easily verifiable that for *ES* (2, 1) and *ES* (2, 2) (that is when n_h= 2, 3) the γ-set contains all the vertices of level 1 only. As we proceed to Case 2 of the algorithm, we consider all n_h of the form 3*m* + 1, *m* = 1, 2, 3…, that is, n_h = 4, 7, 10, 13…. In general, the zero level contains the largest number of nodes. We can see that, each node in level 1, dominates *n*! nodes of level 0. Hence we ignore level 0 and choose level 1 nodes and it effectively reduces the number of nodes in the dominating set. We note that, the nodes chosen in level 1 dominate all the nodes in level 2. Hence we turn our attention to level 3. We note that, each of the $n!$ nodes of level 3 are dominated by a single vertex in level 4. Hence we ignore level 3 nodes, and choose all the nodes in level 4. The choosing of all nodes of level 4 results in level 5 being dominated. Thus recursively choosing all nodes of levels 1, 4, 7…, $(3m + 1)$ … yields a dominating set. In *ES* (2, 3), there are no more hierarchical levels and so level 3 nodes must be included in the dominating set, by all means. The same process holds true for *ES* (2, 6), *ES* (2, 9) …Thus when n_h= 4, 7, 10, 13… we choose levels 1, 4, 7 … *k*.

In Case 3 of the algorithm, we consider all extended star graphs *ES* (*n*, *k*) where n_h is not of the form 4, 7, 10, 13, 16….. that is n_h= 5, 6, 8, 9…. We follow the earlier recursive method of choosing levels 1, 4, 7, … In the earlier case, when n_h= 4, 7, 10, …… we find that we choose all the levels of the form $3m + 1$, and the last level to be chosen is the level *k*, which is of the form $(3m + 1) - 1$. But when $n_h \neq 4, 7, 10, \ldots$, we use the above mentioned recursive method to choose all levels of the form $3m + 1$; $m = 0, 1, 2,\ldots$and the last level to be chosen is the number of the form $3m + 1$, immediately preceding n_h. Hence it is clear that the algorithm gives a dominating set of *ES* (*n*, *k*).

We shall now prove that the above algorithm gives the minimal dominating set of *ES* (*n*, *k*). Let *S* be the dominating set of *ES* (*n*, *k*) obtained by the above algorithm. We use the result [5] that a dominating set *S* is a minimal dominating set if and only if (a) *u* is an isolate of *S* or (b) there exists a node $v \in V - S$ for which $N(v) \cap S = \{u\}$. As per the algorithm, the dominating set *S* imbibes all the nodes of level 1, followed by all the nodes of level 4, level 7 etc., depending on the value of *k*. According to the structure of *ES* (*n*, *k*), the level 0 contains $(n!)^k$ nodes, that is $(n!)^{k-1}$ *n*-stars in level 0. Also none of these *n*-stars are adjacent. The nodes of level 1 dominate all the nodes of level 0. In other words, each node of level 1 dominates $n!$ nodes of an *n*-star. Also the nodes of level 2 are dominated by the nodes of level 1. Hence condition (b) holds for each node *v* belonging to level 1. We argue the case, in the same way for the nodes of level 4, level 7, level 10 and so on.

The level *k* contains a single node, namely P_0. In the cases when level *k* node P_0 is included in *S*, we can see that P_0 is an isolate of *S*, since $N(P_0) \subseteq V - S$, where *V* is the vertex set of *ES* (*n*, *k*). Hence condition (a) holds for level *k*. Thus the set *S* as determined by algorithm 3.1.2 is a minimal dominating set of *ES* (*n*, *k*). Since we use a recursive loop structure in the algorithm, which executes at most $k + 1$ times, we can safely say that the worst case complexity does not exceed *k*. In other words, the complexity of the algorithm is O (*k*). ■

4.1.3 Recursive formula - minimal dominating set of *ES* (2, *k*)

Initially, $\gamma(ES(n, 1)) = 1$; $\gamma(ES(n, 2)) = n!$ for all *n*.

For $k \geq 3$,

$\gamma(ES(2, k)) = 2!\ \gamma(ES(2, k-1)) + 1$ if $k = 3, 6, 9, 12\ldots$

$\gamma(ES(2, k)) = 2!\ \gamma(ES(2, k-1)) - (n! - 1)$ if $k = 4, 7, 10, 13\ldots$

$\gamma(ES(2, k)) = 2!\ \gamma(ES(2, k-1))$ if $k = 5, 8, 11, 14\ldots$

4.2 Connected dominating set of *ES* (*n, k*)

4.2.1 Design of algorithm

The following algorithm initializes S as the connected dominating set of *ES* (*n*, *k*). With the input of k, the algorithm first deals with the trivial case, when $k = 1, 2$. The broader case includes all $k \geq 3$ and the set S imbibes all nodes of the level 1, 2…$k - 1$.

4.2.2 Algorithm to find a connected dominating set of *ES* (*n, k*)

Begin
Input k
$S \leftarrow \varphi$
if $k = 1, 2$
then
$S \leftarrow S \cup \{$All nodes of Level "1"$\}$
break;
if $k \geq 3$
then
$S \leftarrow S \cup \{$All nodes belonging to level 1, 2 … $k - 1\}$
Output S
End

Theorem 4.2. Algorithm 4.2.2 determines the connected dominating set of *ES* (*n*, *k*).

Proof: Let T be the spanning tree of *ES* (*n*, *k*) with ε_T end vertices. Since the end vertices of the spanning tree T are the nodes of level 0 ε_T (*ES* (*n*, *k*)) = $(n!)^k$. It is known that $\gamma_c(G) \leq n - \varepsilon_T(G)$ [7]. Hence γ_c(*ES* (*n*, *k*)) $\leq \sum_{j=0}^{k}(n!)^j - (n!)^k$. In other words, γ_c $(G) \leq \sum_{j=0}^{k-1}(n!)^j$. Hence the connected dominating set S consists of all the nodes belonging to levels 1, 2 …k. Since the level k contains a single node P_0 and its removal does not affect the connectivity of S, the minimal dominating set thus obtained, contains all nodes of level 1, 2 … $k - 1$. If S is the connected dominating set of *ES* (2, *k*), then according to its definition $\langle S \rangle$ is connected. For *ES* (2, 1) and *ES* (3, 1) $S = \{ P_0 \}$ is itself a connected dominating set. For *ES* (2, 2), $S = \{P_1, P_2\}$, which are the nodes belonging to level 1 alone form a connected dominating set.

For example, the connected dominating sets of *ES* (2, 3) and *ES* (2, 4) are illustrated in Figure 5 and 6 respectively.

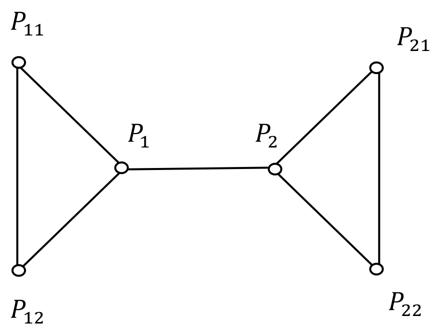

Figure 5: *Connected dominating set of ES* (2, 3)**.**

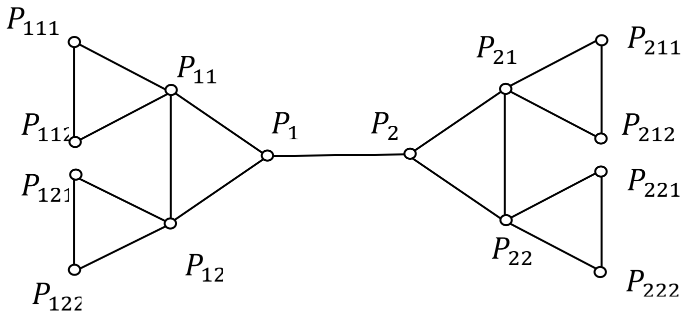

Figure 6: *Connected dominating set of ES (2, 4).*

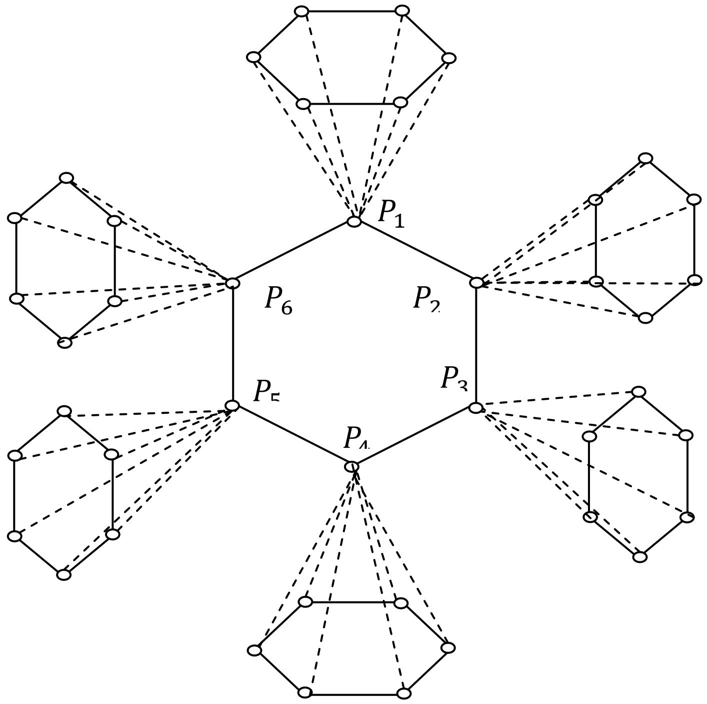

Figure 7: *Connected dominating set of ES (3, 3).*

Hence for $k \geq 3$, the subgraph induced by the dominating set S is a bi-pyramidal structure connected by the bridge P_1P_2. In *ES* (3, 2) also the dominating set *S* consists of nodes of level 1 only and hence they form a hexagon which is a connected structure.

For $k \geq 3$ in *ES* (3, *k*), the subgraph induced by *S* consists of a main 3-star and six 3-stars branching out from each of the six nodes and so on. For example, in *ES* (3, 3), the subgraph induced by *S* is of the following shape. The nodes of level 1 and level 2 of *ES* (3, 3) and the graph that they induce is shown in Figure 7.

Thus the algorithm 4.2.2 determines a minimal connected dominating set of *ES* (*n*, *k*). ■

4.3 Total dominating set of ES (*n, k*)

4.3.1 Design of algorithm

The following algorithm uses the same technique as Algorithm 3.1.2, to find a minimal dominating set of *ES* (*n*, *k*) and then convert it into a total dominating set. We define $n_h = k + 1$ and we divide all values of $n_h = k + 1$ into 3 mutually independent cases, namely

Case 1: When $n_h = 2, 3$

Case 2: When n_h is of the form $3m + 1$or $3m + 2$

Case 3: When n_h is of the form $3m$

All the above mentioned cases use recursion to determine the total dominating set of *ES* (*n*, *k*). Any isolated node that is obtained by applying the technique of Algorithm 3.1.2, is converted into an edge, by including a node from the subsequent level, that is adjacent to the said isolated node.

4.3.2 Algorithm to find the total dominating set of ES (*n, k*)

Begin
Input k
Int n_h, i, t ;
$n_h = k + 1$
$S \leftarrow \varphi$
If $n_h = 2, 3$
  Go to Case 1
Else
  If Remainder ($n_h/3$) = 1or 2
    Go to Case 2
Else
  If Remainder ($n_h/3$) = 0
    Go to Case 3
Case 1:
  if $n_h = 2$
    then
      $S \leftarrow S \cup \{P_0\} \cup$ {Any one node from level "0"}
  break;
  if $n_h = 3$
    then
      $S \leftarrow S \cup$ {All nodes of Level "1"}
  break;
Case 2:
$S \leftarrow S \cup \{P_0\} \cup$ {Any one node of level "$k - 1$"}
 for $i = 0$ to k, $i = i + 1$ do
   $t \leftarrow 3i + 1$;
    while ($k - t \geq 2$))
     do
      $S \leftarrow S \cup$ {All nodes of Level "t "}

break;
Case 3:
for $i = 0$ to k , $i = i + 1$ do
$t \leftarrow 3i + 1$;
while ($k - t \geq 0$)
do
$S \leftarrow S \cup \{$All nodes of Level "t"$\}$
break;
End.

Theorem 4.3. Algorithm 4.3.2 determines a total dominating set of *ES* (n, k) and its complexity is O (k)**.**

Proof: Since the necessary condition for a dominating set S to be a total dominating set is that S has no isolated nodes, we improvise upon algorithm 3.1.2 and we convert every isolated vertex in S into an edge with the above algorithm. The algorithm uses the same structure as the Algorithm 4.1.2 with a minor modification in the body of the for loop structure. Hence we can see that the complexity of the algorithm is not disturbed and remains as O (k). ■

5. Domination parameters of the extended star graph with cross-connections

5.1. Minimal dominating set of *ESC* (*n*, *k*)

5.1.1 Design of algorithm

The following algorithm strives to determine the minimal dominating set of *ESC* (n, k) by dividing all the values of n_h into three mutually independent cases namely the trivial case when (i) $n_h = 3$, case (ii) when n_h is of the form $3m + 1$, $m = 1, 2, 3...$ and finally case (iii) when n_h is not of the form $3m + 1$, $m = 1, 2, 3...$ In each case, there are a series of steps, which when executed gives us the list of levels, whose nodes form the dominating set of the graph *ESC* (n, k).

5.1.2 Algorithm: To Find the dominating set of *ESC* (*n*, *k*)

Begin
Input k
int n_h, i, t, m, r
$n_h \leftarrow k + 1$
$S \leftarrow \varphi$
Case 1:
If $n_h = 2$
$S \leftarrow S \cup \{ P_0, P_0' \}$
If $n_h = 3$
$S \leftarrow S \cup \{$ All nodes of level 1$\} \cup \{$ All nodes of level $2'\}$
break;
Case 2:
If $n_h = 3m + 1$, $m = 1, 2, 3...$

then
$S \leftarrow S \cup \{P_0\}$
for $i = 0$, $i = i + 1$ do
$t \leftarrow 3i + 1$
$r \leftarrow t + 1$
while $(k - t) \geq 2$ do
$S \leftarrow S \cup$ {All nodes of level t}$\cup${All nodes of level r'}
break;

Case 3:

If $n_h \neq 3m + 1$, $m = 1, 2, 3...$
then
for $i = 0$, $i = i + 1$ do
$t \leftarrow 3i + 1$, $r \leftarrow t + 1$
while $(k - t) \geq 1$
$S \leftarrow S \cup$ {All nodes of level "t"} $\cup$ {All nodes of level r'}
if $(k - t) = 0$
then
$S \leftarrow S \cup${All nodes of level "t"} $\cup$ {All nodes of level t'}
break;
Output S
End.

Theorem 5.1. Algorithm 5.1.2 determines a minimal dominating set of *ESC* (n, k).

Proof: The algorithm follows the same technique of Algorithm 3.1.2 to find the minimal dominating set of *ESC* (n, k). We follow the bottom to top approach in building the minimal dominating set of *ESC* (n, k). The *ESC* (n, k) can be visualized as two extended star graphs namely *ES* (n, k) and ES (n, $k - 1$) with cross connections in between them.

The zero level has the greatest number of nodes in *ES* (n, k) and likewise the level $1'$ has the greatest number of nodes in *ES* (n, $k - 1$). Hence we ignore level 0 as well as level $1'$ and choose the nodes of level 1 and $2'$. In the process, level 2 and level $3'$ are dominated. Hence we turn our attention to level 3 and level $4'$. We note that each of the $n!$ nodes in level 3 are dominated by a single node in level 4 and likewise each of the $n!$ nodes in level $4'$ are dominated by a single node in level $5'$ and there are cross connections between level 3 and level $4'$. Hence it suffices to choose all the nodes in level 4 and level $5'$. By choosing level 4 and level $5'$ we clearly see that, level 5 and level $6'$ are dominated. Thus recursively choosing all nodes of levels 1, 4, 7... $(3m + 1)$... as well as levels $2'$, $5'$, $8'$...$(3m + 1) + 1'$, ... yields a dominating set. In *ESC* (n, 3), there are no more hierarchical levels and so level 3 nodes must be included in the dominating set, by all means. The same process holds true for *ESC* (n, 6), *ESC* (n, 9)... Thus when n_h= 4, 7, 10, 13..., in additions to choosing levels 1, 4, 7... $(3m + 1)$ .... we choose the level, namely the single node P_0.

In case 3 of the algorithm, we consider all *ESC* (n, k) where n_h is not of the form 4, 7, 10, 13... that is n_h is of the form 5, 6, 8, 9... In the earlier case, when n_h= 4, 7, 10... we find that we chose all the levels of the form 3m + 1, along with the level k. But in this case, the last level to be chosen is the level of the form 3m + 1, immediately preceding n_h. In the same way, we choose level (3m + 1)$'$, whenever n_h is of the form 3m + 2, m = 1, 2, 3... and the level (3m + 2)$'$when n_h is of the form 3m, m = 1, 2, 3... Hence it is clear that the algorithm gives the dominating set of *ESC* (n, k).

We now prove that the above algorithm gives the minimal dominating set of *ESC* (n, k). Let S be the dominating set of *ESC* (n, k) obtained by the above algorithm. We use the result[5] that a dominating set is a minimal dominating set if and only if (a) u is an isolate of S or (b) there exists a node $v \in V - S$ for which $N(v) \cap S = \{u\}$. As per the algorithm, the dominating set S imbibes all the nodes of level 1, followed by all the nodes of level 4, level 7 etc., depending on the value of k. The nodes of level 2$'$, level 5$'$, level 8$'$ etc. are also imbibed by S.

According to the structure of *ESC* (n, k), the level 0 contains $(n!)^k$ nodes, that is $(n!)^{k-1}n$-stars in level 0. Also none of these n-stars are adjacent. The nodes of level 1 dominate all the nodes of level 0. In other words each node of level 1 dominates $n!$ nodes of n-star. Also the nodes of level 2 are dominated by the nodes of level 1. Hence condition (b) holds for each node v belonging to level 1. We can argue the case, in the same way for the nodes of level 4, level 7, level 10 and so on and also for the level 2$'$, level 5$'$, level 8$'$ and so on.

The level k as well as the level k', contains a single node, namely P_0. In the cases, when level k node P_0 or the level k' node P_0' is included in the dominating set S, we see that P_0 is an isolate of S since $N(P_0) \subseteq V - S$, and also $N(P_0') \subseteq V - S$ where V is the vertex set of *ESC* (n, k). Hence condition (a) holds for level k or level k'whenever it is included in S. Thus as determined by the algorithm, S is a minimal dominating set of *ESC* (n, k). ■

5.2 Connected dominating set of *ESC* (*n, k*)

5.2.1 Design of algorithm

The following algorithm first deals with the trivial case of k = 1 and k = 2 and then uses a recursive loop to include all levels 1, 2 ... k – 1 and 2$'$, 3$'$… (k – 1)$'$. The algorithm ignores both the top and bottom levels but includes all the levels in between them.

5.2.2 Algorithm to find the connected dominating set of *ESC* (*n, k*).

```
Begin
Input k
S ← φ
if k = 1
  then
    S ←S ∪{P0, P0′}∪{ Any one node of level 0}
break;
if k = 2
  then
```

$S \leftarrow S \cup$ { All nodes of level "1"} $\cup$ {All nodes of level $2'$}
break;
for $k \geq 3$ do
$S \leftarrow S \cup$ { All nodes of levels 1, 2 ... $k-1$} $\cup$ {All nodes of levels $2', 3'..(k-1)'$}
Output *S*.
End.

Theorem 5.2. The algorithm 5.2.2 determines a connected dominating set of *ESC* (*n*, *k*).

Proof: We first deal with the trivial case of $k = 1$. The choice of nodes P_0, P_0' and any one of the nodes from level 0 results in the path P_3 that is connected and also dominates both level 0 and level 1 and also the nodes of the *n*-star.

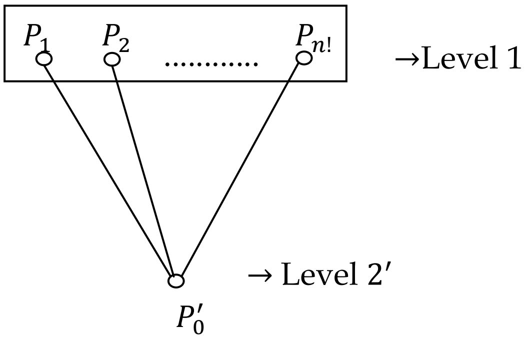

Figure 8: *Connected dominating set of ESC* (*n*, 2).

When $k = 2$, the nodes of level 1 dominates both the level 2 and level 0. In order to dominate both levels$1'$ and $2'$ it is enough to choose P_0' (level $2'$) alone since it dominates all the nodes of level $1'$and there are cross-connections between level $2'$ and level 1. Thus the nodes of level 1 and level $2'$ form a connected dominating set of *ESC* (*n*, 2). Figure 8 shows the connected dominating set of *ESC* (*n*, 2).
When $k > 2$, we choose all the nodes of level 1, 2 … $k-1$ as well as the nodes of levels $2'$, $3'$… k'. Since there are cross-connections between 1 and $2'$, 2 and $3'$ and likewise between $(k-1)$ and k' we construct a connected dominating set that dominates both *ES* (*n*, *k*) and the *ES* (*n*, $k-1$) with which it has cross-connections. ■

5.3 Total dominating set of *ESC* (*n, k*)

5.3.1 Design of algorithm

The following algorithm determines the total dominating set of *ESC* (*n*, *k*) by dividing all the values of $n_h = k + 1$, into 3 mutually independent cases, namely
(i) When $n_h = 2, 3$
(ii) When n_h is of the form $3m + 1$, $m = 1, 2, 3...$
(iii) When n_h is not of the form $3m + 1$, $m = 1, 2, 3...$

In each case, there are a series of steps, which when executed, constructs the lists of levels, whose nodes form the total dominating set of *ESC* (*n*, *k*).

5.3.2 Algorithm to find the total dominating set of *ESC* (*n, k*)

Begin
Input k
int n_h, i, t, m, r
$n_h \leftarrow k + 1$
$S \leftarrow \varphi$
Case 1:
If $n_h = 2$
$S \leftarrow S \cup \{ P_0, P_0' \} \cup \{$ Any one node of level 0$\}$
If $n_h = 3$
$S \leftarrow S \cup \{$All nodes of level 1$\} \cup \{$All nodes of level 2$'\}$
break;
Case 2:
If Remainder (n_h/3) = 1,
then
  $S \leftarrow S \cup \{P_0, P_1\}$
   for i = 0, i = i + 1 do
    $t \leftarrow 3i + 1$
    $r \leftarrow t + 1$
  while $(k - t) \geq 2$ do
  $S \leftarrow S \cup \{$All nodes of level $t\} \cup \{$All nodes of level $r'\}$
  break;
Case 3:
If Remainder (n_h/3) = 0 or 1,
then
 for i = 0, i = i + 1 do
  $t \leftarrow 3i + 1$, $r \leftarrow t + 1$
   while $(k - t) \geq 2$
  $S \leftarrow S \cup \{$All nodes of level "t"$\} \cup \{$All nodes of level $r'\}$
  if $(k - t) = 0$
   then
    $S \leftarrow S \cup \{ P_0, P_0', P_1, P_1' \}$
  else
   if $(k - t) = 1$
    $S \leftarrow S \cup \{P_0', P_1'\} \cup \{$All nodes of level "$t$"$\}$
break;
Output S
End.

Theorem 5.3. Algorithm 5.3.2 determines the total dominating set of *ESC* (*n*, *k*).

Proof: The algorithm 5.3.2 is similar to Algorithm 5.1.2, which constructs a minimal dominating set of *ESC* (*n*, *k*). However, the latter allows isolated nodes, which are converted into edges in algorithm 4.3.2, by choosing a node from the subsequent level. In this case, P_0 and P_0' being the only possibilities of isolated nodes, we choose P_1 and P_1' of levels $k-1$ and $(k-1)'$ respectively. The process ensures that the induced sub graph $\langle S \rangle$ has no isolated nodes and thereby we have constructed a total dominating set, which is also minimal. ■

Conclusion

We have discussed and designed algorithms to determine some of the dominating sets that are relevant to resource allocation in parallel processing network. The above described algorithms can be adapted to work on other parallel processing networks that have a similar structure.

References

[1] S. B. Akers and B. Krishnamurthy, *A group theoretic model for symmetric interconnection networks*, IEEE Transactions on Computer, 38(4), 1989, 555-566.

[2] A. Anto Kinsley, S. Somasundaram and C. Xavier, *Fault- tolerant Hierarchical Network of Star Graphs*, Computer Science and Informatics, Vol. 29, No. 1(1999), 16-21.

[3] S. Arumugam and R. Kala, *Domination Parameters of Hypercubes*, Journal of the Indian Mathematical Society, Vol. 65, Nos. 1- 4 (1998), 31-38.

[4] S. Arumugam and R. Kala, *Domination Parameters of Star graph*, Ars Combinatoria. 44(1996), 93-96.

[5] G. Chartrand and L. Lesniak, *Graphs and Digraphs*, CRC, (2005).

[6] K. Day and A. Tripathi, *A Comparative Study of Topological Properties of Hypercubes and Star Graphs*, IEEE Transactions on parallel and distribution systems 5(1), (1994), 31-38.

[7] Teresa W. Haynes, Stephen T. Hedetniemi, Peter J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker Inc., 1998.

[8] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, *Domination in Graphs- Advanced Topics*, Marcel Dekker, Inc., NewYork, (1997).